A group signature scheme based on the integer factorization and the subgroup discrete logarithm problems

R. Durán Díaz[1], L. Hernández Encinas[2], and J. Muñoz Masqué[2]

[1] Universidad de Alcalá, 28871-Alcalá de Henares, Spain
raul.duran@uah.es
[2] Instituto de Física Aplicada, CSIC, 28006-Madrid, Spain
{luis, jaime}@iec.csic.es

Abstract. Group signature schemes allow a user, belonging to a specific group of users, to sign a message in an anonymous way on behalf of the group. In general, these schemes need the collaboration of a Trusted Third Party which, in case of a dispute, can reveal the identity of the real signer. A new group signature scheme is presented whose security is based on the Integer Factorization Problem (IFP) and on the Subgroup Discrete Logarithm Problem (SDLP).

Key words: Digital signature, Group signature, Public key cryptography

1 Introduction

As it is well-known, there are different protocols to determine digital signatures. In general, these protocols are based on public key cryptosystems [1–3]. The main characteristic of this signature schemes is that each signer has one public key and one private key.

Moreover, the procedures of digital signatures are made more efficient if hash functions are used [4]. The hash functions are public and they allow to sign a digest or hash of the message.

Group signature schemes were proposed by Chaum and van Heyst in 1991 [5]. These schemes permit a signer group to sign a given message such that only a member of the group computes the signature on behalf of the whole group. A Trusted Third Party ($\mathcal{T}$) collaborates in the generation of the keys and is able to reveal the identity of the user who signed the message, if a dispute arises.

The main characteristics defining the group signatures are the following:

1. Only a member of the signer group signs the message.
2. The receiver of the message can verify that the signature of the message was generated by a member of the signer group, but he cannot determine which member of the group was the signer.
3. If a dispute arises, it is possible to open the signature in order to determine who was the actual signer of the message.

Group signatures can be understood as an extension of credential authentication and membership authentication schemes. In the first schemes, a user proves that he belongs to a specific group [6]; whereas in the second ones, a member of a group can convince a verifier that he belongs to that group without revealing him his identity [7, 8].

There exist several proposals for group signatures, which use a number of cryptographic primitives. Some of these proposals need a Trusted Third Party (TTP), $\mathcal{T}$, at least for the initialization process. Other schemes, however, allow any user to create the group he chooses to belong to.

As a general rule, group signatures make use of schemes whose security is based on computationally-intractable mathematical problems [9–11]. Typically, such problems are the Integer Factorization Problem (IFP) and the Discrete Logarithm Problem (DLP).

Nevertheless, most of these protocols show some limitations. For example, the schemes described in [12–14] have a security problem [15]. Moreover, the security of the schemes presented in [16, 17] is tested under artificial and unlikely conditions [18].

The proposed group signature scheme presented here guarantees that a true group signature is generated for a given message. Moreover, the scheme improves existing protocols in terms of user friendliness, computational efficiency, time and band-width saving. Moreover, this proposal verifies the properties required for group signature schemes: Only a group member can validly sign a document or message. The signed-message receiver is able to verify that the signature is a valid group signature, *i.e.*, it has been carried out by one legitimate member of the group. However, the receiver will not be able to determine which particular group member actually signed the message. Finally, if required (in case of a dispute, for example) it is possible to disclose the signer, *i.e.*, to reveal which user actually signed the message.

The rest of this paper is organized as follows: In section 2 a group signature scheme based on the Integer Factorization and Subgroup Discrete Logarithm Problems is proposed. In section 3, the main properties of the new scheme are shown. The security analysis of the proposal is performed in section 4, and finally, the conclusions are presented in section 5.

2 A group signature scheme based on IFP and SDLP

In this section we propose a group signature scheme for which a randomly chosen member of a given group signs a document, on behalf of the whole group, making use of his private key. The verifier of the signature checks whether or not the signature corresponds to one of them, using the public key that all the members of the group share. Moreover, the verifier will not be able to decide who was the original signer.

Let $G = \{U_1, U_2, \ldots, U_t\}$ be the signer group and let $\mathcal{T}$ be the Trusted Third Party.

2.1 Setup phase

In this phase, $\mathcal{T}$ generates its pre-key, the public key shared by the group, as well as helps the members of G to generate their private keys [19].

Pre-key generation $\mathcal{T}$ generates its pre-key as follows:

1. $\mathcal{T}$ chooses two large primes p and q, such that
$$\begin{aligned} p &= u_1 \cdot r \cdot p_1 + 1, \\ q &= u_2 \cdot r \cdot q_1 + 1, \end{aligned}$$
where r, p_1, q_1 are prime numbers, $u_1, u_2 \in \mathbb{Z}$ with $\gcd(u_1, u_2) = 2$, that is, $u_1 = 2v_1$, $u_2 = 2v_2$, and $\gcd(v_1, v_2) = 1$.
In order to guarantee the security of the scheme, the bitlength of r is selected so that the Subgroup Discrete Logarithm Problem (SDLP) of order r in $\mathbb{Z}_n^*$ be computationally infeasible.
2. $\mathcal{T}$ computes
$$\begin{aligned} n &= p \cdot q, \\ \phi(n) &= (p-1)(q-1) = u_1 \cdot u_2 \cdot r^2 \cdot p_1 \cdot q_1, \\ \lambda(n) &= \operatorname{lcm}(p-1, q-1) = \frac{\phi(n)}{\gcd(p-1, q-1)} = 2v_1 \cdot v_2 \cdot r \cdot p_1 \cdot q_1, \end{aligned}$$
where $\phi(n)$ is the Euler function, $\lambda(n)$ is the Carmichael function, and lcm represents the least common multiple.
Then, $\mathcal{T}$ selects an element $\alpha \in \mathbb{Z}_n^*$ with multiplicative order r modulo n, such that
$$\gcd(\alpha, \phi(n)) = \gcd(\alpha, u_1 \cdot u_2 \cdot r^2 \cdot p_1 \cdot q_1) = 1.$$
Note that this element, α, can be efficiently computed as $\mathcal{T}$ knows the factorization of n and consequently it knows $\phi(n)$ and $\lambda(n)$ [19, Lemma 3.1].
We denote by S_r the subgroup of $\mathbb{Z}_n^*$ generated by α.
3. $\mathcal{T}$ generates a secret random number $s \in \mathbb{Z}_r^*$ and determines
$$\beta = \alpha^s \pmod{n}. \tag{1}$$
4. $\mathcal{T}$ publishes the values (α, r, β, n); whereas it keeps secret the values of (p, q, s).

With the previous hypothesis, the security of $\mathcal{T}$'s secret, s, is based on the Integer Factorization Problem (IFP) and on the Subgroup Discrete Logarithm Problem (SDLP).

Key generation In order to determine the private keys of the members of G, $\mathcal{T}$ computes its private key and the public key which will be shared by all the signers of G.

To do this, $\mathcal{T}$ generates four random numbers $a_0, b_0, c_0, d_0 \in \mathbb{Z}_r^*$ as its private key and determines the shared public key for G by computing

$$\left.\begin{array}{l} P = \alpha^{a_0} \cdot \beta^{b_0} \pmod{n} \\ Q = \beta^{c_0} \cdot \alpha^{d_0} \pmod{n} \end{array}\right\} \tag{2}$$

From (2), we have

$$\begin{aligned} P &\equiv \alpha^{a_0} (\alpha^s)^{b_0} \pmod{n} \equiv \alpha^{a_0 + s \cdot b_0} \pmod{n}, \\ Q &\equiv (\alpha^s)^{c_0} \alpha^{d_0} \pmod{n} \equiv \alpha^{s \cdot c_0 + d_0} \pmod{n}. \end{aligned}$$

Hence, $P, Q \in S_r$, that is, there exist integers $h, k \in \mathbb{Z}_r$ such that

$$\left.\begin{array}{l} h = (a_0 + s \cdot b_0) \pmod{r} \\ k = (s \cdot c_0 + d_0) \pmod{r} \end{array}\right\} \tag{3}$$

In order to guarantee that $\mathcal{T}$ cannot impersonate any user of G, an interactive session between each user U_i and $\mathcal{T}$ is necessary to determine the private key of U_i, $1 \leq i \leq t$. Hence, the following interactive protocol is developed:

1. U_i generates two secret integers $b_i, d_i \in \mathbb{Z}_r$ at random and sends to $\mathcal{T}$ the values of $\alpha^{b_i}, \alpha^{d_i}$, in a secure way for protecting both secret integers.
2. $\mathcal{T}$ computes

$$\begin{aligned} A_i &= \alpha^h \cdot (\alpha^{b_i})^{-s} \pmod{n} = \alpha^{a_i}, \\ C_i &= \alpha^k \cdot (\alpha^{d_i})^{-1} \pmod{n} = \beta^{c_i}. \end{aligned}$$

   From (3), $\mathcal{T}$ can compute A_i, C_i since it knows h, k, α^{b_i}, and α^{d_i}, but it cannot compute a_i, c_i because it cannot solve the SDLP. Then $\mathcal{T}$ sends to U_i the values of A_i, C_i by using a secure channel.
3. The private key of U_i is the set (b_i, d_i, A_i, C_i). Note that for U_i is also impossible to compute the values of a_i, c_i.

Remark. Note that $\mathcal{T}$ knows two values of the U_i's private, A_i, C_i, but it is impossible for it to know the rest of that key. Moreover, for both U_i and $\mathcal{T}$ it is impossible to compute the values a_i, c_i because they are protected by the SDLP.

Key verification For verifying the pre-key of $\mathcal{T}$, each members of the signer group, U_i, $1 \leq i \leq t$, must check

$$\begin{aligned} \alpha &\not\equiv 1 \pmod{n}, \\ \alpha^r &\equiv 1 \pmod{n}. \end{aligned}$$

Moreover, each signer, U_i, $1 \leq i \leq t$, must verify that his private key corresponds to the shared public key, i.e., must check if it holds:

$$P \equiv A_i \cdot \beta^{b_i} \pmod{n}, \tag{4}$$

$$Q \equiv C_i \cdot \alpha^{d_i} \pmod{n}. \tag{5}$$

In fact:

$$\begin{aligned} A_i \cdot \beta^{b_i} \pmod{n} &\equiv \alpha^{a_i} \cdot \beta^{b_i} = \alpha^{a_i + s \cdot b_i} = \alpha^h = P, \\ C_i \cdot \alpha^{d_i} \pmod{n} &\equiv \beta^{c_i} \cdot \alpha^{d_i} = \alpha^{s \cdot c_i + d_i} = \alpha^k = Q. \end{aligned}$$

2.2 Group signature generation

Let M be the message to be signed by a member of G. We can assume that after computing its hash value (by using, for example, a public hash function from the SHA-2 family), we have $\mathfrak{h}(M) = m$. For signing M on behalf of the group G, a random and anonymous member of G is chosen, for example, U_i. Next, U_i does the following.

1. U_i generates a secret integer $\lambda_i \in \mathbb{Z}_r$ at random. This value must be generated each time a message is signed.
2. U_i determines his signature, (F_i, G_i, H_i), for M, computing the following values:
$$\left.\begin{array}{l} F_i = A_i \cdot C_i^m \cdot \alpha^{\lambda_i} \pmod{n} \\ G_i = \beta^{b_i} \cdot (\alpha^{d_i})^m \cdot \alpha^{-\lambda_i} \pmod{n} \\ H_i = \mathfrak{h}(\alpha^{\lambda_i}) \end{array}\right\} \tag{6}$$
3. Finally, $\mathcal{T}$ publishes the group signature for the message M: $(F, G, H) = (F_i, G_i, H_i)$.

Remark. Nobody can impersonate the user U_i because he is the only one knowing the values b_i, d_i, and λ_i.

2.3 Group signature verification

Let (F, G, H) be a group signature of G for the message M. In order to verify this signature, any verifier knowing the public key of the group G, (P, Q), can check that

$$P \cdot Q^m \equiv F \cdot G \pmod{n}. \tag{7}$$

The equation (7) can be immediately justified from expressions (4)-(6) as follows:

$$\begin{aligned} F \cdot G \pmod{n} &\equiv A_i \cdot C_i^m \cdot \alpha^{\lambda_i} \cdot \beta^{b_i} \cdot \alpha^{m \cdot d_i} \cdot \alpha^{-\lambda_i} \pmod{n} \\ &= A_i \cdot \beta^{b_i} \cdot C_i^m \cdot \alpha^{m \cdot d_i} \\ &= P \cdot Q^m. \end{aligned}$$

3 Properties of the new scheme

The proposed scheme has the following properties:

1. All the operations involved in the different phases described in the previous paragraphs can be efficiently computed in polynomial time.
2. Despite $\mathcal{T}$ knows part of U_i's private key, it cannot forge the signature determined by U_i as the signer has generated at random the value: λ_i. Nevertheless, it can generate a valid group signature.
3. The verifier is only able to test whether the signature was generated by a member of the signer group and it is not able to ascertain the identity of the actual signer.
4. In case of dispute, $\mathcal{T}$ can disclose the signer since it knows part of the private key of each member of G.
   In fact, as $\mathcal{T}$ knows the values of A_i and C_i of the signer U_i, by using the equations in (6) defining the group signature, it can compute
   $$\frac{F}{A_i \cdot C_i^m} \pmod{n} \equiv \frac{A_i \cdot C_i^m \cdot \alpha^{\lambda_i}}{A_i \cdot C_i^m} = \alpha^{\lambda_i}.$$
   Then, $\mathcal{T}$ can prove, without the collaboration of U_i, that
   $$\mathfrak{h}\left(\frac{F}{A_i \cdot C_i^m} \pmod{n}\right) = \mathfrak{h}(\alpha^{\lambda_i}) = H_i.$$

4 Security analysis

Moreover, the scheme is secure as no member of G, say U_i, knowing only his own private key, (b_i, d_i, A_i, C_i), and the shared public key, $(P = \alpha^{a_0 + s \cdot b_0}, Q = \alpha^{s \cdot c_0 + d_0})$, can determine neither the secret value s of $\mathcal{T}$, nor its private key (a_0, b_0, c_0, d_0).

In fact, determining s from α and $\beta \equiv \alpha^s \pmod{n}$, see formula (1), means solving the discrete logarithm problem in the subgroup S_r, of order r generated by α, which is impossible as the size of r was chosen such that the SDLP was unfeasible to solve, and moreover, the factorization of n is infeasible as well.

Moreover, the private key of $\mathcal{T}$ was generated at random and it is only known that it verifies the equation (2), but computing any of the values of this key implies solving the DLP in $\mathbb{Z}_n^*$.

It is also impossible for any U_i to determine the values of $h = a_i + s \cdot b_i$, and $k = s \cdot c_i + d_i$, as he only knows $b_i, d_i, \alpha^{a_i}, \beta^{c_i}$. In all cases, it is necessary to solve a discrete logarithm problem.

Furthermore, two members of G, say U_i and U_j, could conspire and try to compute any of the secret values of $\mathcal{T}$: $s, h, k, a_0, b_0, c_0, d_0$, or generate a false signature for the group. To carry out any of these attacks, both could generate their signatures for a message, say (F_i, G_i, H_i) and (F_j, G_j, H_j), respectively. Then, from the verification identity (7), they have

$$F_i \cdot G_i \pmod{n} \equiv F_j \cdot G_j = P \cdot Q^m.$$

Hence, they obtain

$$A_i \cdot C_i^m \cdot \beta^{b_i} \cdot \alpha^{m \cdot d_i} \equiv A_j \cdot C_j^m \cdot \beta^{b_j} \cdot \alpha^{m \cdot d_j} \pmod{n},$$

or equivalently,

$$\alpha^{a_i} \cdot \beta^{m \cdot c_i} \cdot \beta^{b_i} \cdot \alpha^{m \cdot d_i} \equiv \alpha^{a_j} \cdot \beta^{m \cdot c_j} \cdot \beta^{b_j} \cdot \alpha^{m \cdot d_j} \pmod{n},$$

and as α has order r modulo n, it results

$$(a_i + m \cdot d_i) + s(b_i + m \cdot c_i) \equiv (a_j + m \cdot d_j) + s(b_j + m \cdot c_j) \pmod{r}.$$

that is, they could obtain

$$s \equiv (a_i - a_j + m(d_i - d_j)) \cdot (b_j - b_i + m(c_j - c_i))^{-1} \pmod{r}.$$

Nevertheless, none of them know the values of a_i, a_j, c_i, c_j, so they cannot compute s.

Finally, nobody is able to forge a group signature for the message M without this fact being detected and proved by $\mathcal{T}$. In fact, a forger could know the public key, (P, Q), the message, M, its hash, m, and the values (α, r, β, n). From these data, the forger can choose an element $\widetilde{G} \in S_r$, determine the value

$$\widetilde{F} = P \cdot Q^m \cdot \widetilde{G}^{-1} \pmod{n},$$

and publish the set $(\widetilde{F}, \widetilde{G}, \widetilde{H})$, for a hash value $\widetilde{H}$, as a group signature for the message M, that passes the verification equation (7).

Nevertheless, $\mathcal{T}$ can prove that this group signature is a forgery by computing

$$\overline{H}_i = \mathfrak{h}\left(\frac{\widetilde{F}}{A_i \cdot C_i^m} \pmod{n} \right), \quad 1 \leq i \leq t,$$

and showing that $\overline{H}_i \neq \widetilde{H}$, $\forall i$.

5 Conclusions

A new group signature scheme has been proposed. The security of the scheme is based on two difficult problems from Number Theory: Integer factorization and subgroup discrete logarithms (and the DLP in the key generation).

The scheme verifies the properties required for general group signature schemes. Any single member of the signer group is able to sign the message. The receiver of the message can verify that the signature of the message was generated by a actual member of the signer group, but he cannot determine which member of the group was the signer. If a dispute arises, a Trusted Third Party can open the signature and determine who was the signer of the message.

The group signature scheme is efficient since the computations only require polynomial time and moreover it is secure against conspiracy attacks and against forgery.

Acknowledgment. This work has been partially supported by the "Fundación Memoria D. Samuel Solórzano Barruso" under the Project FS/7-2010.

References

1. ElGamal, T.: A public-key cryptosystem and a signature scheme based on discrete logarithm. IEEE Trans. Inform. Theory 31, 469–472 (1985)
2. Menezes, A., van Oorschot P., Vanstone, S.: Handbook of applied cryptography. CRC Press, Boca Raton, Florida (1997)
3. Rivest, R.L., Shamir, A., Adleman, L.: A method for obtaining digital signatures and public-key cryptosystems. Comm. ACM 21, 120–126 (1978)
4. National Institute of Standards and Technology: Secure Hash Standard (SHS). Federal Information Processing Standard Publication 180-2 (2002)
5. Chaum, D., van Heyst, E.: Group signatures, In: Davies, D.W. (ed.) EUROCRYPT 1991. LNCS, vol. 547, pp. 257–265. Springer, Heidelberg (1991)
6. Chaum, D.: Showing credentials without identification. In: Pichler, F. (ed.) EUROCRYPT 1985. LNCS, vol. 219, pp. 241–244. Springer, Heidelberg (1985)
7. Ohta, K., Okamoto, T., Koyama, K.: Membership authentication for hierarchical multigroup using the extended Fiat-Shamir scheme. In: Damgård, I.B. (ed.) EUROCRYPT 1990. LNCS, vol. 473, pp. 446–457. Springer, Heidelberg (1990)
8. Shizuya, H., Koyama, S., Itoh, T.: Demonstrating possession without revelating factors and its applications. In: Seberry, J., Pieprzyk, J.P. (eds.) AUSCRYPT 1990. LNCS, vol. 453, pp. 273–293. Springer, Heidelberg (1990)
9. Bresson, E., Stern, J.: Efficient revocation in group signature. In Kim, K.C. (ed.) PKC 2001. LNCS, vol. 1992, pp. 190–206. Springer, Heidelberg (2001)
10. Camenish, J., Michels, M.: Separability and efficiency for generic group signature schemes. In: Wiener, M. (ed.) CRYPTO 1999. LNCS, vol. 1666, pp. 413–430. Springer, Heidelberg (1999)
11. Camenish, J., Stadler, M.: Efficient group signature schemes for large groups. In: CRYPTO 1997. LNCS, vol. 1296, pp. 410–424. Springer, Heidelberg (1997)
12. Ateniese, G., Camenish, J., Joye, M., Tsudik, G.: A practical and provably secure coalition-resistant group signature scheme. In: Bellare, M. (ed.) CRYPTO 2000. LNCS, vol. 1880, pp. 255–270. Springer, Heidelberg (2000)
13. Ateniese, G., de Medeiros, B.: Efficient group signatures without trapdoors. In: Laih, C.S. (ed.) ASIACRYPT 2003. LNCS, vol. 2894, pp. 246–268. Springer, Heidelberg (2003)
14. Nguyen, L., Safavi-Naini, R.: Efficient and provably secure trapdoor-free group signature schemes from bilinear pairings. In. Lee, P.J. (ed.) ASIACRYPT 2004. LNCS, vol. 3329, pp. 89–102. Springer, Heidelberg (2004)
15. Shoup, V., Gennaro, R.: Securing threshold cryptosystems against chosen ciphertext attack. J. Cryptology 15, 2, 75–96 (2002)
16. Boneh, D., Boiyen, X., Shacham, H.: Short group signatures. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 41–55. Springer, Heidelberg (2004)
17. Camenich, J., Lysyanskaya, A.: Signature schemes and anonymous credentials from bilinear maps. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 56–72. Springer, Heidelberg (2004)
18. Boneh, D., Boyen, X.: Short signatures without random oracles. In: Cachin, C., Camenish, J. (eds.) EUROCRYPT 2004. LNCS, vol 3027, pp. 56–73. Springer, Heidelberg (2004)
19. Susilo, W.: Short fail-stop signature scheme based on factorization and discrete logarithm assumptions. Theor. Comput. Sci. 410, 736–744 (2009)